I/O

July-September 2014

A quarterly newsletter for LANL’s HPC user community

Trinity Issue

In this issue

LA-UR-14-27892

Things you should know............

Standard service features

The Integrated Computing Network (ICN) Consulting Team provides user support on a wide variety of HPC topics:

- Programming, languages, debugging
- Parallel computing
- HPC and Unix operating systems, utilities, libraries
- Unix / Linux scripting
- Archival storage: High Performance Storage System (HPSS), General Parallel File System (GPFS)
- Desktop backup (TSM), storage, file transfer, network
- Mercury: Cross-network file transfer service
- HPC infrastructure in both the Open and Secure

Service hours

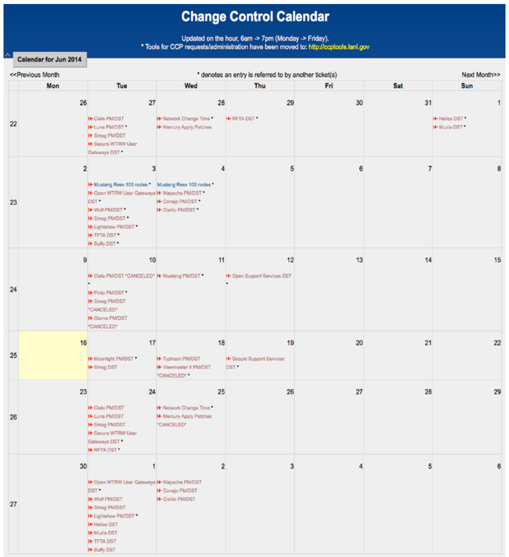
Monday through Friday, 8 a.m. - 12 p.m. and 1 p.m. - 5 p.m
After-hours/Urgent support-can transfer to 7x24 operations desk

Phone #
505-665-4444 opt 3

Email
consult@lanl.gov

Documentation
http://hpc.lanl.gov

HPC Change Control Calendar
http://ccp.lanl.gov



Change Control Calendar

Updated on the hour, 6am -> 7pm (Monday -> Friday).
* Tools for CCP requests/administration have been moved to: http://ccptools.lanl.gov

Calendar for Jun 2014

<<Previous Month   * denotes an entry is referred to by another ticket(s)   Next Month>>

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
22	26	27 Cielo PM/DST Luna PM/DST * Smog PM/DST Secure WTRW User Gateways DST *	28 Network Change Time * Mercury Apply Patches	29 RFTA DST *	30	31	1 Helios DST * Muzia DST *
23	2	3 Mustang Resv 100 nodes * Open WTRW User Gateways DST * Wolf PM/DST * Smog PM/DST * Lightshow PM/DST * TFTA DST * Buffy DST *	4 Mustang Resv 100 nodes * Mapache PM/DST * Conejo PM/DST * Cielito PM/DST *	5	6	7	8
24	9	10 Cielo PM/DST *CANCELED* * Pinto PM/DST * Smog PM/DST *CANCELED* Glome PM/DST *CANCELED*	11 Mustang PM/DST *	12 Open Support Services DST *	13	14	15
25	16	17 Moonlight PM/DST * Smog DST	18 Typhoon PM/DST Viewmaster II PM/DST *CANCELED* *	19 Secure Support Services DST *	20	21	22
26	23	24 Cielo PM/DST Luna PM/DST Smog PM/DST Secure WTRW User Gateways DST * RFTA DST *	25 Network Change Time * Mercury Apply Patches *CANCELED*	26	27	28	29
27	30	1 Open WTRW User Gateways DST * Wolf PM/DST Smog PM/DST Lightshow PM/DST * Helios DST Muzia DST TFTA DST Buffy DST	2 Mapache PM/DST Conejo PM/DST Cielito PM/DST	3	4	5	6

Consultants' Corner

Best Practices for Archiving Data from HPC Filesystems

On all of our HPC platforms, we provide large, volatile parallel filesystems as scratch space for your applications. This is a shared resource among all users that is generally large and fast for parallel I/O. We offer scratch filesystems only as temporary space until you can use your data and then remove it to make room for other user applications. Since many of our users need to retain this data and not lose it, we provide offline archival storage for that purpose.

Secure Restricted (i.e. Red, Classified) Network

Our Integrated Computing Network (ICN) offers a special cluster of File Transfer Agents (FTAs), designed to offload user data movement and file transfer traffic from large HPC platforms by providing high bandwidth that is independent of any compute cluster. You can access the FTAs with a simple `ssh rfta-fe` command from your workstation or from the Red Network gateway, red-wtrw.lanl.gov.

The FTAs can perform high-speed parallel file transfers between the offline HPSS archive and the online HPC scratch filesystems. The easiest way to exploit high-bandwidth is with these commands on the FTAs. This is an interactive example that you can run on the rfta-fe node:

```
msub -I -l nodes=4,walltime=…
psi store -R --cond /
scratch6/$USER:/hpss/$USER/red_scratch6
```

What Does This Do?

The `msub` command starts an interactive job for you, it requests an allocation of FTA nodes within an interactive Moab job.
Here is what happens with the psi command line:

- The store action will examine your Moab allocation, i.e. the number of nodes you requested with `msub`.
- `psi` will then open-up parallel streams on your allocated nodes in-between HPSS and /scratch6 and begin the parallel file transfer automatically.
- The `--cond` option will only store files to HPSS that are newer on /scratch6. This allows it to pick-up where it left off in case of an interruption, and it avoids redundant stores of the same files.

The transfer will continue until it finishes moving data, or else it will terminate your FTA Moab job if it runs out of time.

You can also use this technique with a batch FTA job. We provide a simple job script as an example here: http://hpc.lanl.gov/fta_home. If using batch mode, you can submit a dependent job that will launch afterward and continue the same file transfers.

For more information on HPSS, see:
http://hpc.lanl.gov/hpss

Open Collaborative (i.e. Turquoise, Unclassified) Network

In the Turquoise network, we now provide Campaign Storage for medium-term offline storage of your working data. This gives you more time to archive data over to GPFS and preserve it long-term while you free-up online scratch space for other user applications.

We mount the Campaign Storage on new Turquoise nodes, Campaign File Transfer Agents (CFTAs). These are accessible from their front-end: `cfta-fe`, and you can expect an average transfer bandwidth of 1 GB/s to and from Campaign Storage. As usual, you can only reach the CFTAs from the Turquoise gateway, wtrw.lanl.gov. You can obtain highest speeds from fewer, large files -- the nodes are throttled by many, small files. When moving data to and from Campaign Storage, we recommend using the `/usr/local/bin/pfcp` parallel file copy tool which provides greater file transfer bandwidth.

You can request Campaign Storage space from Institutional Computing:
http://int.lanl.gov/org/padste/adtsc/institutional-computing.shtml
(Unclassified Networks only)

For more information on Campaign Storage, see:
http://hpc.lanl.gov/turquoise_filesystems#CampaignStorage

For a permanent archival service, use our General Parallel File System (GPFS):
http://hpc.lanl.gov/turquoise_archive
This allows you to preserve your important data for long-term so you can delete it from scratch space.

Unclassified Protected (i.e. Yellow) Network
The Yellow Network sees light HPC activity compared to the Red and Turquoise networks, and we do not provide Yellow FTA services. Our only option for archiving your important data is to run the psi command from any Yellow Network front-end node to the offline Yellow HPSS. You can achieve maximum bandwidth by avoiding Moonlight since it sees the heaviest activity among the Yellow front-ends. Otherwise, we offer no parallel file transfer capability in our Yellow Network.

Note: to access unclassified hpc.lanl.gov websites from outside of LANL, see instructions here:
http://www.lanl.gov/projects/computing/web_hpc.html

Consultants
left to right, back to front
Ben Santos, Hal Marshall, Riley Arnaudville, Rob Derrick
Giovanni Cone, Rob Cunningham and David Kratzer

Software and Tool News

Programming Environments and Runtime Team Announcements:

Process improvement has been the center of the software team's attention, in an effort to create a more robust and stable working environment for scientists running on LANL Clusters. Over the quarter, we will be collecting feedback from code teams about their requirements and release cycles, and weaving that into the process definitions we're formalizing. The goal is to streamline installation processes, freeing up staff-members to focus on usage and specialization in different aspects of Scientific Computing. As always, we welcome your input! Please feel free to notify us at ptools_team@lanl.gov of any special requirements or enhancement recommendations to the programming environment that will improve your HPC experience at LANL.

New Changes to the Processes:
Improving communication with our Customers is our first priority. We are now handling Programming Environment changes in a more

formalized and predictable manner. The change notification / implementation process is as follows:
1. The first and third Thursday of every Month is reserved for changes to the production environment.
2. On those days, we will send notification of changes that were made to Production Software, as well as a list of proposed changes for the next maintenance day.
3. We can add or subtract friendly-testing modulefiles at any time and we will announce a list of those changes in the same notification to users.

If you find that a proposed change will adversely affect your work-flow or milestones, please don't hesitate to notify us via the ICN Consulting Office. These are proposed changes only, and we will remain flexible to accommodate our customers needs.

Featured Software Products -Intel®:
Intel® Cluster Studio XE 2013 is now licensed and installed in friendly-testing space on all production clusters. This suite includes comprehensive support for HPC hybrid parallel programming. Cluster Studio includes Intel® Trace Analyzer and Collector for MPI communication and correctness analysis. A simple tutorial for Intel's® Trace Analyzer can be found at:
https://software.intel.com/en-us/itac_9.0_analyzing_app

For shared and hybrid application analysis, the Vtune Amplifier/Inspector tools are available. Advisor XE is a product that supports analysis, design, prototyping and tuning of threading designs, exploration of threading options without code disruption as well as scaling support for threaded applications with higher core counts. You can find more information:
https://software.intel.com/en-us/intel-advisor-xe/

Additionally, Intel® Cluster Studio XE 2013 provides Intel MPI with MPI-3 specification support on multiple fabrics. The Intel MPI reference manual can be found at:
https://software.intel.com/sites/products/documentation/hpc/mpi/linux/reference_manual.pdf.

The 2015 Intel Compiler has been released, and will soon be available for friendly-testing. The compiler includes:
New standards support:
- OpenMP 4.0
- Full Fortran 2003
- C++11
- Added Fortran 2008 BLOCK construct feature

Full release notes can be found at:
https://software.intel.com/en-us/articles/intel-parallel-studio-xe-2015-cluster-edition-initial-release-readme

The compiler is part of Intel's Parallel Studio (formerly Cluster Studio) tool suite and includes analysis tools such as VTune Amplifier for performance profiling, Trace analyzer, and the latest version of Intel-MPI. All of these products will be available in friendly-testing, soon.

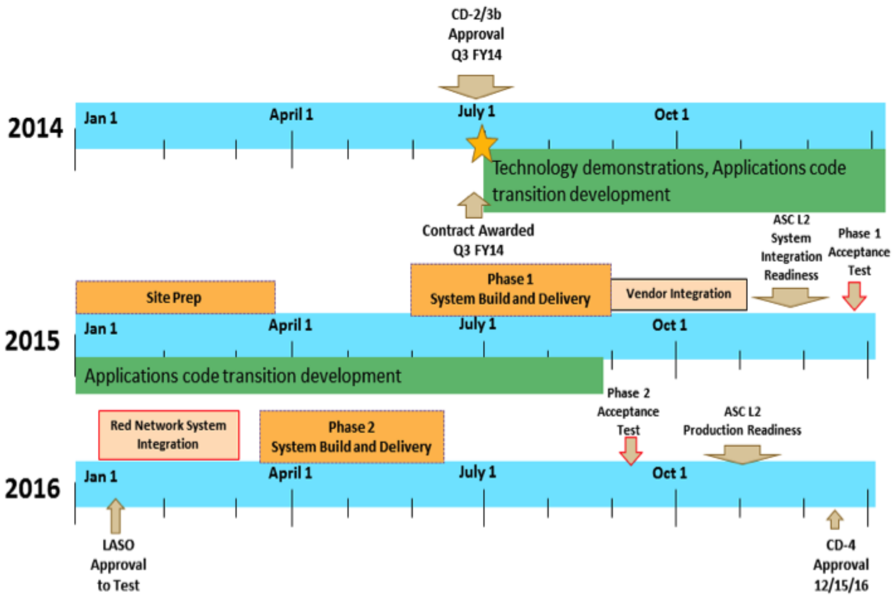
Programming Environments and Runtime Team
left to right
David Gunter, Riley Arnaudville, Jennifer Green,
David Shrader, Giovanni Cone, Marti Hill,
and Jorge Roman

Machines News

What is Trinity?

Introduction

The Trinity computer platform will be the first advanced technology system (ATS) for the NNSA tri-Lab Advanced Simulation and Computing Program (ASC). It will satisfy the mission need for more capable platforms. Trinity is designed to support the largest, most demanding ASC applications and increases in geometric and physics fidelities while satisfying analysts' time-to-solution expectations. While based on mature Cray XC30 architecture, Trinity also introduces new architectural features, including Burst Buffer(BB) storage nodes, advanced power management (APM) system software, and the Intel Knights Landing (KNL) processor. Trinity will be installed in phases, as shown in the schedule.


CD-2/3b Approval Q3 FY14
2014
Jan 1
April 1
July 1
Oct 1
Technology demonstrations, Applications code transition development
Contract Awarded Q3 FY14
ASC L2 System Integration Readiness
Phase 1 Acceptance Test
Site Prep
Phase 1 System Build and Delivery
Vendor Integration
2015
Jan 1
April 1
July 1
Oct 1
Applications code transition development
Phase 2 Acceptance Test
ASC L2 Production Readiness
Red Network System Integration
Phase 2 System Build and Delivery
2016
Jan 1
April 1
July 1
Oct 1
LASO Approval to Test
CD-4 Approval 12/15/16

New Architectures

Trinity is enabling new architectures in a production computing environment. Here are summaries of three of these features:

- Burst Buffer, tightly coupled solid state storage, enables improved time to solution efficiencies for checkpoint and restart file I/O and data analytics.
- Advanced power management to enable measurement and control at the system, node and component levels, allowing exploration of application performance per watt and reducing total cost of ownership.
- Trinity will be a single system with both Intel Haswell and Knights Landing (KNL) processors. The Haswell partition satisfies FY15 mission needs (well suited to existing codes). The KNL partition, to be delivered in FY16, will result in a system significantly more capable than current platforms and provides the application developers with an attractive next generation target.

Minimizing risk with the Cray XC30

The Cray XC30 architecture minimizes system software risk and provides a mature high-speed interconnect.

Center of Excellence for Application Transition

Trinity will benefit from the Center of Excellence (CoE) for Application Transition, a collaboration of the NNSA tri-labs, Cray, and Intel. The Center is essential for ensuring key ASC applications will successfully port to perform on the Trinity architecture.

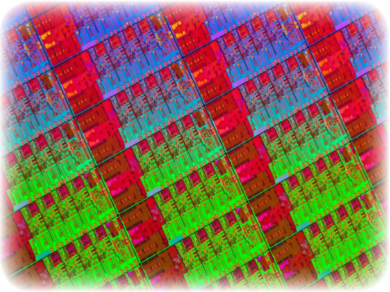
Water-cooled

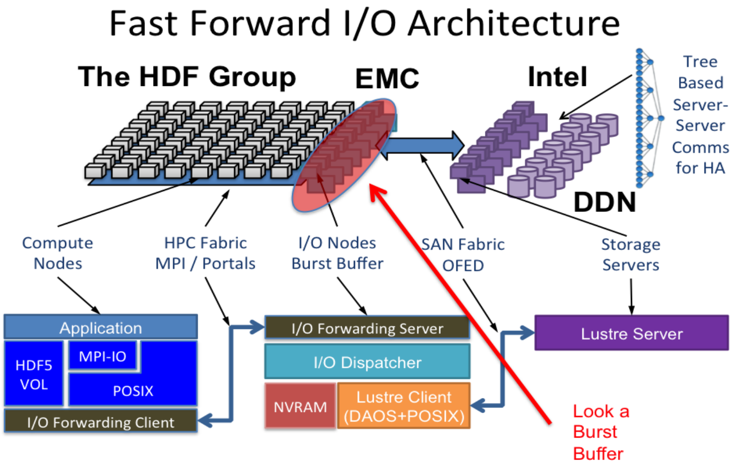
Trinity will be water cooled. Liquid is more efficient than air in moving the heat generated by blades and other components. This efficiency also means that racks can be more densely packed with a smaller footprint.

User challenges

Trinity's architecture will introduce new challenges for code teams: including the transition from multi-core to many-core, a high-speed on-chip memory subsystem, wider SIMD/vector units. Although the KNL processor is a higher risk as a new technology, it offers a reasonable path for code teams to transition to many-core architecture.

Looking down the path, Trinity is expected to foster a competitive environment and influence next-generation architectures in the HPC industry.





The I/O portion of the DOE Fast Forward program funds industry to accelerate advances in data flow.

Origin of Burst Buffers

The word "burst" was ubiquitous at the SC13 conference, and you have probably read by now that the Trinity platform will include burst buffers as part of its design. What you may not know is that the term burst buffer was coined here at LANL by Gary Grider, HPC Division Leader, several years ago. More importantly, you may not understand the role of a burst buffer, why it's necessary, and how it may evolve or even eventually disappear again altogether.

The red network HPSS archive contains data generated as early as the 1960s. Since then, the basic execution model leading to the kind of data stored has not changed very much; in fact, it's a variant of the Von Neumann model: Load, Compute, Store. Simulations are run, physical state is saved and/or examined from time to time, and both input and output data are archived. A lot of attention has been paid to Load (if we think of it as including creation of applications in a given build environment) and Compute, but the difficulties inherent in the Store portion of this model are mounting quickly. Trinity will be the first platform to do something other than grow basic data movement and storage infrastructure to address these problems.

Over the years HPC Division has been able to predict future archival growth from computer system main memory—archival data fluctuates between one and three main memories per month. Cielo, Typhoon, and Luna together have roughly 350 TB of main memory, and the HPSS archive has grown on average 1.8 times that per month over the last three years, 22PB of the current 42PB in the archive. Trinity's main memory is over 2PB. Twice that would be 4PB archived per month, 144PB over three years. Provisioning the HPSS archive to both take in data at that rate and store it would cost tens of millions of dollars the ASC program currently uses for other purposes, such as salaries. Suppose that volume of data did go into the archive—getting any of it out again is yet more difficult. Reading out a half a petabyte restart dump at current read rates would take over 20 continuous days. So that is Reason One the current execution model has to change, and the burst buffer can play a role.

On the other end of the execution model as it applies to storage, applications write restart dumps to a file system to hedge against machine failure. The RFP for Trinity specified that the system must be able to write out 80 percent of system memory in 20 minutes or less. That's about 1.4 TB/s. Trinity's file system will just about be able to do that, running flat out, but as anyone who has paid attention to file I/O performance on a busy supercomputer knows, there are a lot of things in the way of “flat out”. We’ll call this Reason Two for having a burst buffer.

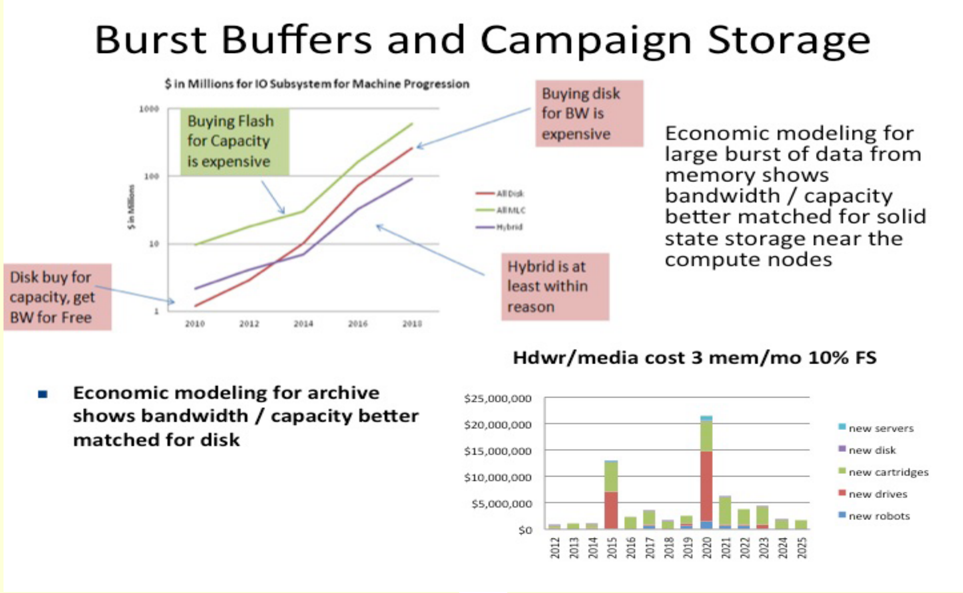
Reason Two—accelerating I/O, in particular for restart dumps—is the most well-understood purpose of a burst buffer and the source of the name: Burst buffers in this sense are flash-based devices to which applications can send bursts of data for eventual migration to the file system. Trinity’s burst buffer infrastructure consists of over 3PB of storage allowing a write rate of over 3TB/s, which should alleviate wait time for defensive I/O and allow the current execution model to function for a while longer.

Reasons One and Two for burst buffers are in pretty direct opposition—extending the current restart dump model generates a lot of data we would then not like to store. This conflict exposes the conflict inherent in an AT system clearly—extend the current execution model with burst buffers while advancing to a next-generation execution model . . . with burst buffers. To see how, think of burst buffers not as a part of the computing storage hierarchy, but as part of the memory hierarchy. There may be opportunities for analysis or computational steering taking place in the larger memory of the burst buffer that lead to more efficient data output from computer to storage. How that would work in practice for Trinity is the subject of a lot of current work in CCS, XCP, HPC, Cray, and elsewhere, but the essential goal is an updated execution model for computation and analysis leading to less output of full physical state, higher end-to-end performance, and better user productivity.

e exploitation of the burst buffer for scientific data efficiency is work that likely continues for the lifetime of the system.

The burst buffer is just one element in a changing HPC storage infrastructure. All elements of the HPC storage infrastructure, from burst buffer to archive, are due for evaluation and potential change—see, for example, an article in the June, 2014 issue of the I/O newsletter about the first deployment of what we call campaign storage. Campaign storage leverages technologies evolving in the commercial cloud storage market, which dwarfs HPC storage, to more efficiently and reliably store large volumes of data.

Looking ahead to ATS-3, scheduled for 2020, it’s not clear what the role of what we now call the burst buffer will become. In its role as part of the storage hierarchy, storage vendors may incorporate solid state technologies directly into the products they offer. Technology trends may drastically change the economic balance between spinning and solid state media. The memory hierarchy will likely be deeper and 3D memory technologies may have matured. The ASC program will have more detailed choices to make for ATS-3, as well as funding to deploy to help shape key system characteristics to support ASC application code environments. Via burst buffers or other technologies, the ATS family of machines will continue to balance the needs of the then-current scientific execution model and the need to advance that model into its next generation.


Burst Buffers and Campaign Storage

Buying Flash for Capacity is expensive

Buying disk for BW is expensive

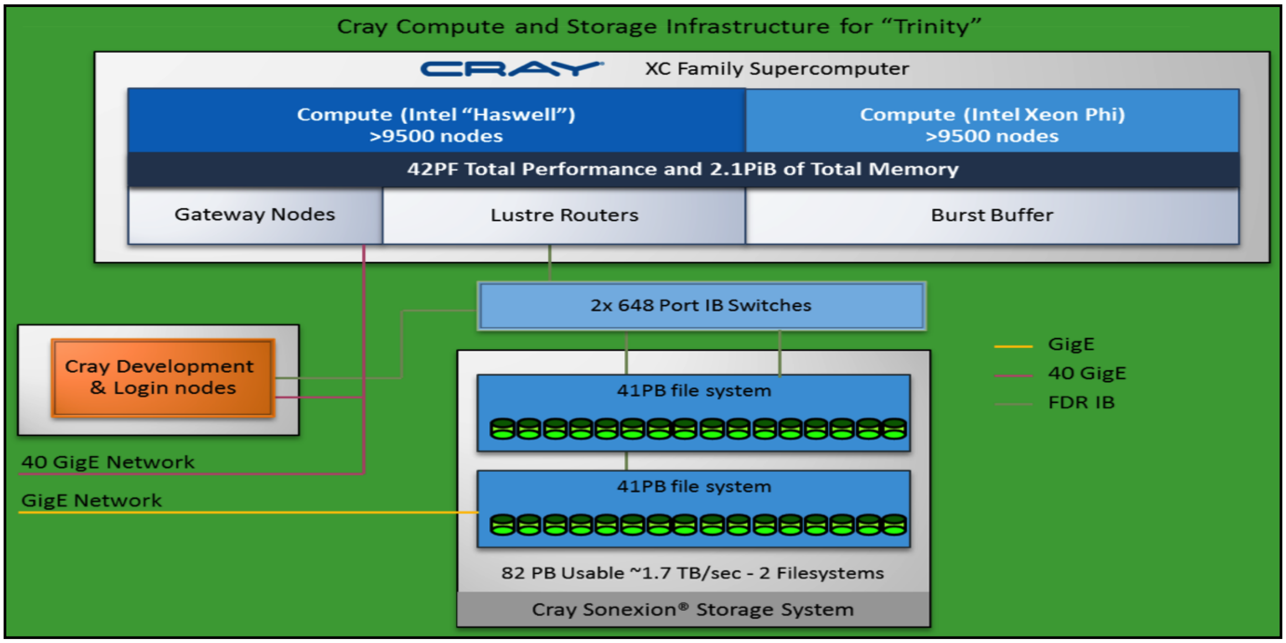
Disk buy for capacity, get BW for Free

Hybrid is at least within reason

Economic modeling for large burst of data from memory shows bandwidth / capacity better matched for solid state storage near the compute nodes

- Economic modeling for archive shows bandwidth / capacity better matched for disk

Hdwr/media cost 3 mem/mo 10% FS


Cray Compute and Storage Infrastructure for "Trinity"
CRAY
XC Family Supercomputer
Compute (Intel "Haswell")
>9500 nodes
Compute (Intel Xeon Phi)
>9500 nodes
42PF Total Performance and 2.1PiB of Total Memory
Gateway Nodes
Lustre Routers
Burst Buffer
2x 648 Port IB Switches
Cray Development & Login nodes
GigE
40 GigE
FDR IB
41PB file system
41PB file system
82 PB Usable ~1.7 TB/sec - 2 Filesystems
Cray Sonexion® Storage System
40 GigE Network
GigE Network

HPC- Behind the Scenes

How did we arrive at Trinity?

As recently as 2011, the national ASC platform strategy defined three kinds of computing platforms: Capacity (e.g. Luna), Capability (e.g. Cielo), and Advanced Architecture (e.g. Roadrunner). For years the merits of the various types of systems have been debated, with some arguing for more capacity at the expense of anything else and others arguing that failing to field advanced architecture systems leaves the program vulnerable to executing a model that will eventually simply fail to work on future computing architectures. One outcome of this debate was the streamlining of the platforms model to two categories: Commodity Technology Systems (CTS) are essentially the same as the former Capacity category, while the other two categories have been merged into Advanced Technology Systems (ATS). Trinity will be the first AT system fielded in the NNSA complex.

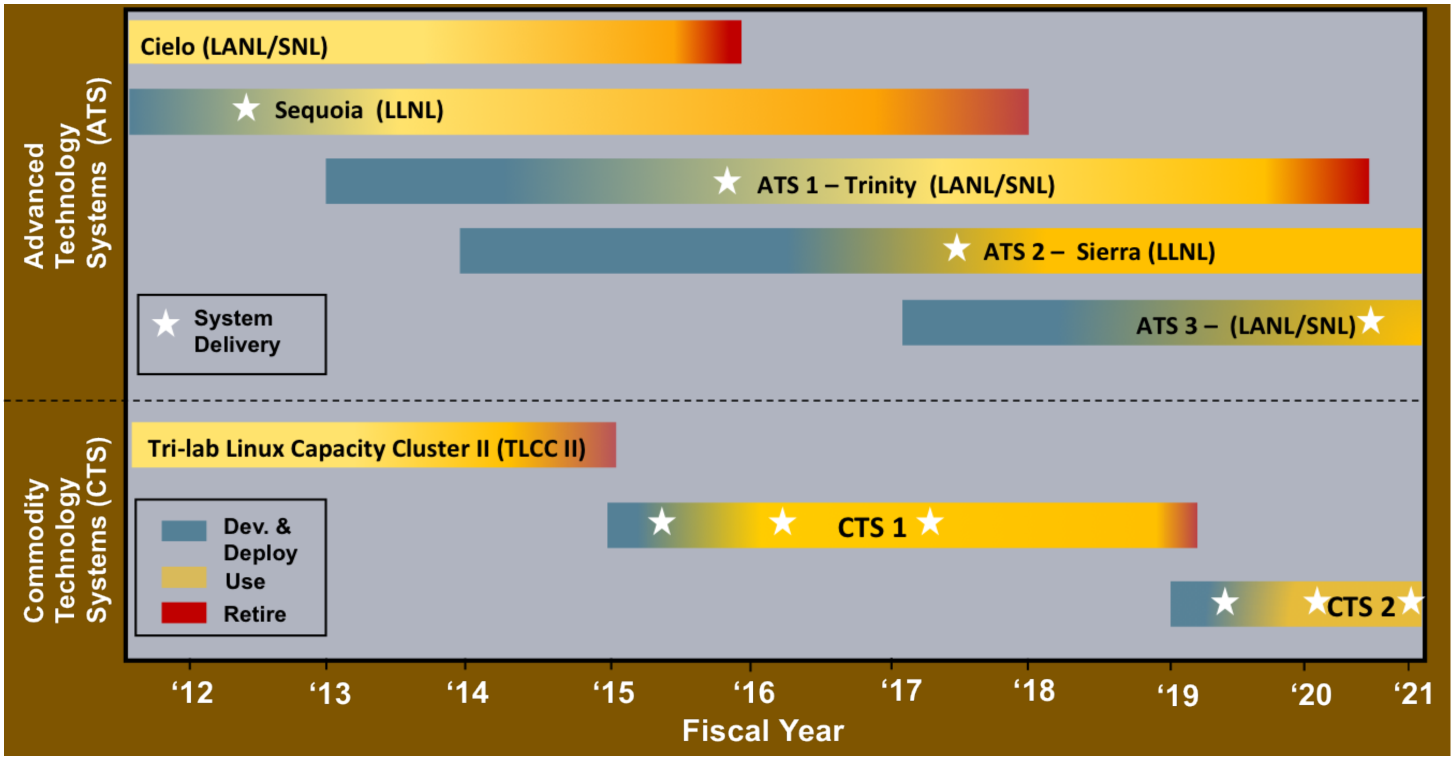
The merging of Advanced Architecture and Capability systems into one Advanced Technology System presented some immediate and obvious difficulties for the specification of the Trinity system. Such a system would have to significantly outperform Cielo, yet allow current applications to run more or less as written and function within a limited power envelope. It would have to employ new technology encouraging the evolution of the ASC scientific execution model, but must essentially be guaranteed to function as designed at the largest scales. LLNL's Sequoia, in contrast, is a capability system, but one with uncertainty quantification as its stated mission—Sequoia's requirement is to run many jobs. Another constraint on Trinity, one that exists for all computer acquisitions, is what computing elements are available and supported in the general timeframe of the acquisition. As an example, about ten years ago HPC personnel were questioned closely at a user forum about the deployment of dual-core compute nodes when it was clear prior to deployment that current applications couldn't efficiently use the second core. The brief answer to that complaint was that single-core nodes were no longer being produced.

For Trinity, too, only certain configurations of computing power were available in the time frame required by the ASC program. Similar dynamics apply to all supercomputer subsystems—memory, interconnect, storage architecture (see the sidebar on Burst Buffers), so the challenge for the ASC program was to prioritize competing constraints, identify technical subsystems for which vendors would have legitimate technology choices in the timeframe required, carefully and completely describe the priorities and constraints, and judge vendor proposals on how well they met those specifications.

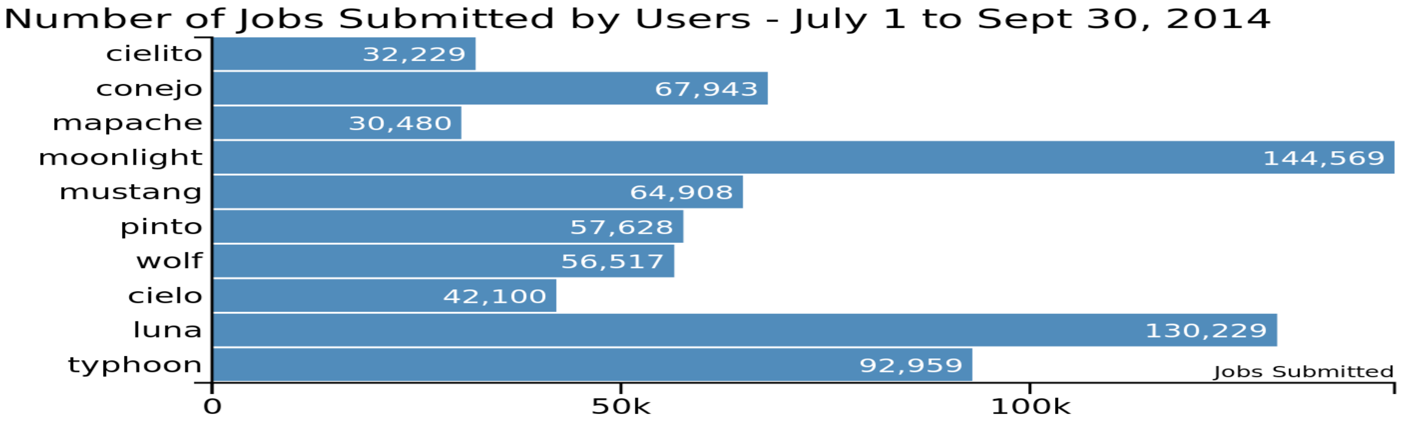
An AT system is necessarily a bridge between immediate performance and future capability. Trinity's Request for Proposal (RFP) captured this need in a number of ways. In an era of decreasing memory per core, for instance, the Trinity RFP specified a minimum of 2PB of main memory to ensure the ability to run large-scale simulations. Supercomputer RFPs typically specify a level of computational performance expressed in floating point operations per second (flops). Trinity's RFP did not. Rather, it expressed performance in terms of full time to solution of a carefully selected suite of representative applications and benchmarks at large scale (2/3 full system size). Flops can be calculated from a processor clock speed and multipliers, while time to solution is a far more complex matter.

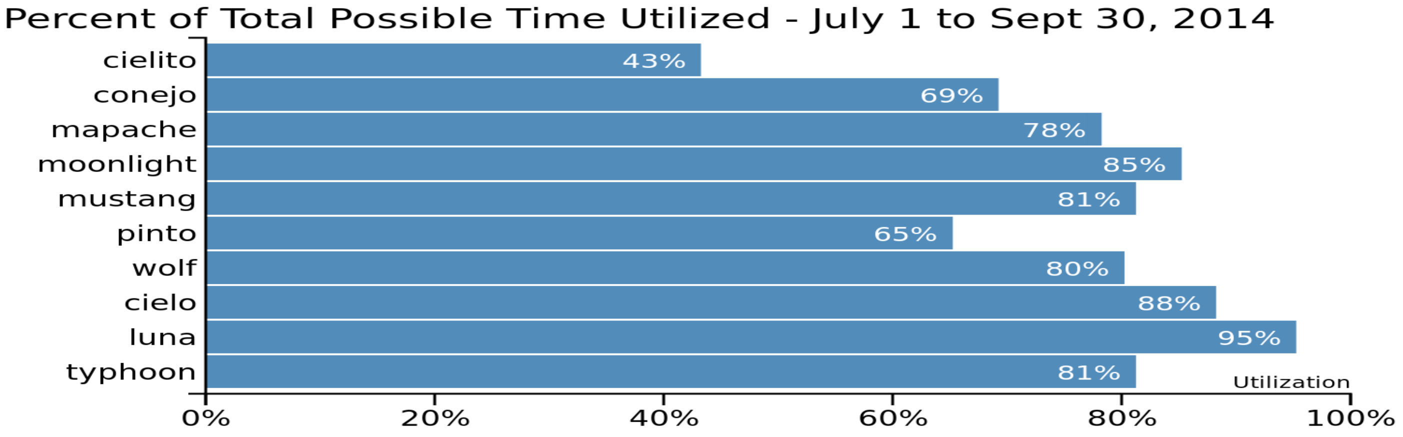
This approach put extra pressure on vendors to provide credible figures and extra pressure on the Trinity team to validate those figures. The RFP also specified caps on power consumption and energy-efficient means for power delivery (480V power) and cooling (warm water cooling). LANL's Sanitary Effluent Reclamation Facility can provide up to 88 millions gallons of water for cooling annually, but with power costs at roughly $1M per megawatt annually and water consumption a perennial concern, budgets of all kinds come into play. Finally, as befits such a bridge system, Trinity's RFP made provisions for ongoing work on advanced power management techniques and burst buffer management software (see sidebar) for the lifetime of the Trinity system.

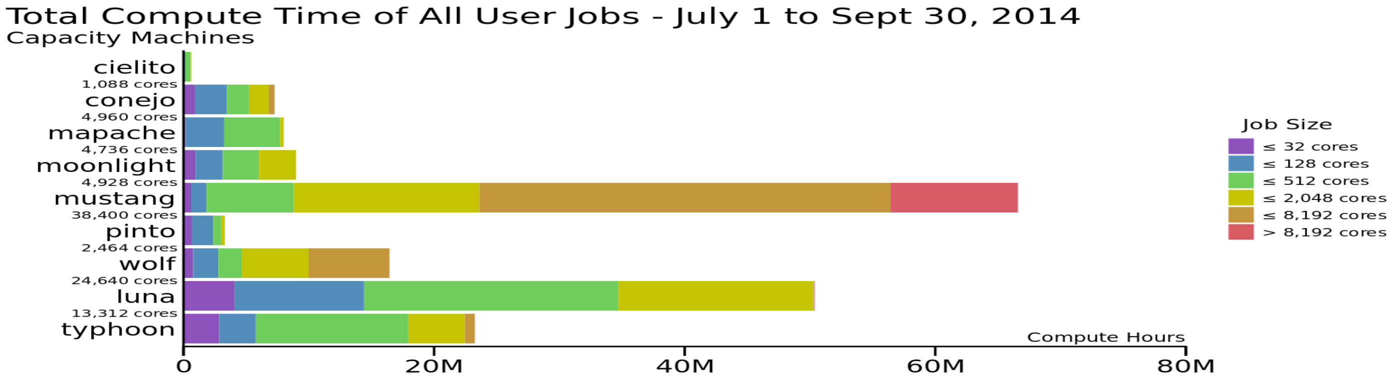
Careful consideration of vendor responses to the RFP and subsequent negotiations with Cray, the selected vendor, have resulted in a system designed to meet the dual goals of an Advanced Technology System: deliver large-scale simulation science upon deployment and position the ASC program to make increasingly effective use of future computing architectures.



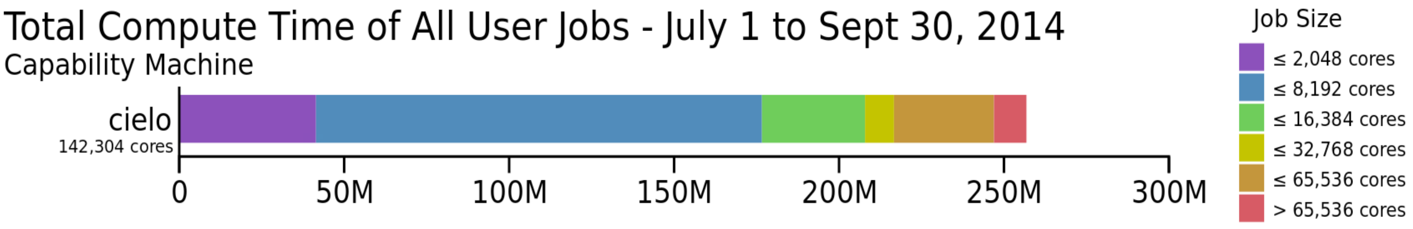
Quarterly Statistics


Number of Jobs Submitted by Users - July 1 to Sept 30, 2014


Percent of Total Possible Time Utilized - July 1 to Sept 30, 2014


Total Compute Time of All User Jobs - July 1 to Sept 30, 2014

Capacity Machines


Total Compute Time of All User Jobs - July 1 to Sept 30, 2014

Capability Machine

Current Machines- A snapshot in time

Name (Program[1])	Processor	OS	Total Compute Nodes	CPU cores per Node/ Total CPUs	Memory per compute Node/Total Memory	Interconnect	Peak (TFlop/s)	Storage
Secure Restricted Network (Red)								
Cielo (ASC)	AMD Magny-Cours	SLES-based CLE and CNL	8,894 nodes	16/142,304	32 GB/297 TB[5]	3D Torus	1,370	10 PB Lustre
Luna TLCC2 (ASC)	Intel Xeon Sandybridge	Linux (Chaos)	1540 nodes	16/24,640	32 GB/49 TB	Qlogic InfiniBand Fat-Tree	513	3.7 PB Panasas
Typhoon (ASC)	AMD Magny-Cours	Linux (Chaos)	416 nodes	32/13,312	64 GB/26.6 TB	Voltaire InfiniBand Fat-Tree	106	3.7 PB Panasas
Open Collaborative Network (Turquoise)								
Cielito (ASC)	AMD Magny-Cours	SLES-based CLE and CNL	68 nodes	16/1088	32 GB/2.3 TB[5]	3D Torus	10.4	344 TB Lustre
Conejo (IC)	Intel Xeon x5550	Linux (Chaos)	620 nodes	8/4960	24 GB/4.9 TB	Mellanox Infiniband Fat-Tree	52.8	1.8 PB Panasas
Lightshow[3] (ASC)	Intel Xeon	Linux (Chaos)	16 nodes	12/192	966 GB/1.5 TB	Mellanox Infiniband Fat-Tree	4.0	1.8 PB Panasas
Mapache (ASC)	Intel Xeon x5550	Linux (Chaos)	592 nodes	8/4736	24 GB/14.2 TB	Mellanox Infiniband Fat-Tree	50.4	1.8 PB Panasas
Moonlight TLCC2[3] (ASC)	Intel Xeon E5-2670 + NVida Tesla M2090	Linux (Chaos)	308 nodes	16/4,928 + GPUs	32 GB/9.86 TB	Qlogic Infiniband Fat-Tree	488	1.8 PB Panasas
Mustang (IC)	AMD Opteron 6176	Linux (Chaos)	1,600 nodes	24/38,400	64 GB/102 TB	Mellanox Infiniband at-Tree	353	1.8 PB Panasas
Pinto TLCC2[3] (IC)	Intel Xeon E5-2670	Linux (Chaos)	154 nodes	16/2464	32 GB/4.9 TB	Qlogic Infiniband Fat-Tree	51.3	1.8 PB Panasas
Wolf TLCC2[3] (IC)	Intel Xeon E5-2670	Linux (Chaos)	616 nodes	16/9856	64 GB/39.4 TB	Qlogic Infiniband Fat-Tree	205	1.8 PB Panasas

[1] Programs: IC=Institutional Computing, ASC=Advanced Simulation and Computing, R=Recharge

[3] TLCC = TriLab Linux Capacity Cluster; 2 = 2nd Generation

[5] Cielo has 372 viz nodes with 64GB memory each

[6] Cielito has 4 viz nodes with 64GB memorv each